DRDC Toronto No. CR 2005-083

Development and Documentation of the Software to Control the Noise Simulation Facility at DRDC Toronto

Élaboration et documentation du logiciel servant à contrôler l'installation de simulation de bruit de RDDC Toronto.

Prepared by Garry Dunn

Trellis Consulting
9 Holly Meadow Road
Barrie, Ontario
L4N 8S1
905-302-7273

Contract #W7711-017769/001/TOR

Scientific Authority: Dr. Sharon M. Abel

Group Leader, Communications
Human Factors Research and Engineering Section
DRDC Toronto
1133 Sheppard Avenue West
Toronto, Ontario
M3M 3B9
Tel: 416-635-2037

December 31, 2004

© HER MAJESTY THE QUEEN IN RIGHT OF CANADA (2004)
As represented by the Minister of National Defence

© SA MAJESTE LA REINE EN DROITE DUE CANADA (2004)
Defense Nationale Canada

Abstract

This document discusses the software used to control the Noise Simulation Facility at DRDC Toronto. Contained within are areas discussing the use of the software by both the administrator of the system as well as the experimenters themselves. The final section discusses how a programmer would expand the system to include more procedures.

Résumé

Ce document traite du logiciel utilisé pour contrôler l'installation de simulation de bruit située à RDDC Toronto. Il comprend des parties qui traitent de l'utilisation du logiciel aussi bien par l'administrateur de système que par les expérimentateurs eux-mêmes. La dernière section examine la façon dont un programmeur pourrait étendre le système pour inclure d'autres procédures.

Executive Summary

The software used to control the Noise Simulation Facility (NSF) at DRDC Toronto is necessarily complex. This document attempts to describe the software system and how it is used.

The first topic discussed concerns the administrative level of the software. At this level, an administrator can create protocols, and specify the various parameters for the protocols. It is intended to be used by the scientists setting up new experiments in the NSF.

The next level of the software is used during the day to day operations of the NSF. It is used to run subjects through the various protocols which have been set up. In general, only subject information is required at this stage. The software system knows the parameters for the subject's session, based on what the administrator has configured for the protocol.

The final component of this document discusses how a programmer would add new procedures to the software system. The system was designed with modularity and expandability in mind so the addition of new procedures would be made easier.

105-05778/

Sommaire analytique

Le logiciel servant à contrôler l'installation de simulation de bruit (NSF) à RDDC Toronto est nécessairement complexe. Ce document essaie de décrire le système logiciel et la manière dont on l'utilise.

Le premier sujet traite les questions au niveau administratif du logiciel. À ce niveau, l'administrateur peut créer des protocoles dont il détermine les différents paramètres. Ce niveau est destiné à l'usage des scientifiques qui lancent de nouvelles expériences dans la NSF.

Le niveau suivant est utilisé pendant les opérations courantes de la NSF. Il sert à exécuter les différents protocoles établis sur les sujets. En général, seuls les renseignements concernant les sujets sont requis à ce stade. Le système logiciel connaît les paramètres nécessaires à la séance des sujets, selon ce que l'administrateur a configuré pour le protocole.

Le dernier élément du document traite de la façon dont le programmeur ajouterait des nouvelles procédures au système logiciel. Le système a été conçu en ayant en vue la modularité et l'extensibilité afin de faciliter l'addition de nouvelles procédures.

Table of Contents

Abstract ii
Résumé iii
Executive Summary iv
Sommaire analytique v
Introduction 7
1 Terms 8
1.1 Protocol 8
1.2 Procedure 8
1.3 Session 8
1.4 Trial 8
2 Directory Structure 9
2.1 /NSF 9
2.2 /SOURCE 9
2.3 /CONFIG 9
2.4 /MRT 9
2.5 /HRTFS 9
2.6 /EXPER 9
3 Using The Software 10
3.1 Administrator Access 10
3.2 Experimenter Access 10
4 Software Detail 11
4.1 Creating New Procedures 11

Introduction

This document describe the structure of the software used in the new program developed for use in the Noise Simulation Facility (NSF) at DRDC Toronto. The software structure is designed to be flexible, expandable and modular. Great effort has gone into creating a system which is easy to add expand. The expansion of the system will rely on a programmer, but this cannot be avoided. The entire system relies on software to generate all sounds going into the NSF. First some history on the NSF and its uses.

The NSF is used to create life-like sound fields. Typically these sound fields are simulations of various military environments (ex: inside helicopters, airplanes, tanks, etc...). The noise levels are generally quite high, yet humans are expected to perform tasks in this environment. The simulated environment gives experimenters a controlled space in which they can perform experiments on subjects, in sound fields, without actually being in the environment. It allows for greater control over parameters of interest to the experimenter.

In the past, the NSF was used simply to create the sound field. The system is setup up to allow for a continuous noise field to be presented. In many situations this is sufficient, however, there are other situations where it is of interest to change the sound field dynamically. This was not possible with the original system.

To make matters more interesting, most of the experiments which took place in the NSF were very labour intensive for the experimenter. It generally required a great deal of paperwork for the experimenter to properly maintain records of what happened in the NSF.

In 1999 an MRT/MLP combination was developed on a computer which proved very useful. It was the first real computerization of an experiment in the NSF and was certainly a good step forward. However, the MRT/MLP was still limited in its use since it required a great deal of knowledge and software expertise to use it. Each experimental idea to be tested required a re-write of the software to suit the purposes of the new experiment. This was not the ideal situation.

The idea behind this new version of software is to eliminate the need for reprogramming the system each time a new experimental protocol comes along. In this vein it is hoped that future experiments can be setup using a set of well defined parameters for each of the various procedures the NSF knows about.

The system is being developed with the future in mind. The structure is simple, yet flexible. This allows for future procedures to be added.

1 Terms

For the purpose of this document and the software being described, terminology will be laid out now so it is clear to the reader as to the topic being discussed.

1.1 Protocol

The term protocol is used to describe an experiment. Protocols are based on a procedure and define how an experiment is run. A protocol might include using the MRT/MLP procedure with 50 presentations, followed by 50 verification presentations.

1.2 Procedure

A procedure is one of the possible test that may be performed using the NSF software. Procedures are programmed and setup for the experimenters to choose from. Examples of a Procedure are the MRT/MLP combination.

1.3 Session

A session is one sitting for a subject of a protocol. A protocol may or may not have multiple sessions.

1.4 Trial

A trial is one elemental task for the subject to perform. In the NSF, this is typically hearing of a particular sound. A session is generally made up of many trials.

2 Directory Structure

This section deals with the directory structure used for the NSF software. Extensive use of folders keeps the system organized.

2.1 /NSF

This is the top level of the program and contains all the software, protocols and results. Basically everything to do with the software is contained under this directory. All further directories listed are relative to this top level.

2.2 /SOURCE

This directory contains all the source, objects and related build files to create the NSF program.

2.3 /CONFIG

This directory contains various configuration files used in the NSF program. These configuration files are maintained by the experimenter/administrator of the system.

2.4 /MRT

This directory contains the word lists and sound files used in the MRT procedures. There are 2 male and 2 female talkers to choose from. The files for each of the talkers are contained within subdirectories of this directory. Those subdirectories are simply called M1, M2, F1 and F2.

2.5 /HRTFS

This directory contains the filters which define the HRTFs used in creating spatialized sounds. There are separate subdirectories for each different HRTF.

2.6 /EXPER

This directory is used to store all the configuration files for each experiment (protocol). The software maintains all the files found in this directory. Each protocol is given its own directory within this one. Subject directories are created beneath this directory for each experiment.

All information contained within this area is stored in standard ASCII files. This makes analysis of the completed experiment simple.

3 Using The Software

This section deals with using the software, at the administrative level, the experimenter level and the subject level. All three levels work together to run an experiment.

3.1 Administrator Access

Before any experiments can be run, they must first be defined. The act of defining an experiment takes place at the administrative level. To invoke the admin level, the user must use the following command:

```
nsf /admin
```

This will give the user special privileges to add/delete/modify protocols in the system. The administrator creates protocols based on pre-programmed procedures in the system. Each of the various parameters for each procedure as chosen by the administrator. Once all parameters have been specified, the protocol is considered 'defined'. The protocol will now be selectable by the experimenter in the main program.

In order to maintain some organization, the administrator has the ability 'deactivate' a protocol. This does not delete any information, it merely removes the protocol from the list available to the experimenter. This allows old, unused protocols to be removed from the list of available protocols. The idea here is that once a protocol is completed, it can be deactivated and no longer viewed by the experimenter. The data, however, remains in the system for closer examination.

The system creates a new directory for each protocol. All information pertaining to that protocol is kept within that directory.

3.2 Experimenter Access

For normal, everyday use the experimenter runs the program with the command:

```
nsf
```

The experimenter is presented a list of the currently active protocols. The experimenter selects the letter associated with the protocol to be run. The behaviour of the software from this point on varies depending on the protocol selected. It is most likely that the experimenter will be asked for such things as:

- Subject name
- Subject gender

The procedure defined by the administrator for this protocol will then be undertaken. Remember: the protocol has already been defined by the administrator so the procedure's parameters (like number of repetitions, talker, frequencies of interested, etc.) have all been defined. It is basically a matter of starting the procedure for the subject.

At this point, the system also takes care of different sessions (if applicable to this procedure). This concept may be difficult to understand in writing, but once it is witnessed on the system, it is easy to see how the system has been developed. The idea is that each experiment is defined by the administrator. When a subject arrives, only basic information is required from the subject. The system knows the rest.

4 Software Detail

This section gives a more detailed look at how the software is constructed. It is meant to be a reference for anyone wishing to change the software. It is intended to be read by a programmer who wishes to have more detail of how the software does what it does. The software source is fully documented and is always the ultimate reference pertaining to how it works, however this section will give a good overview of what can be expected to be found in the software source.

The software can be considered to be a state machine. The mainline of the software merely presents the subject with a list of currently active protocols. When the experimenter asks for a protocol, the mainline calls the function pertaining to that protocol, along with appropriate parameters which tell that protocol how to run. The software has many routines which the various protocols have at their disposal. Included functions are:

- Linked lists
- Menus and their selection
- Log file creation
- Functions to control the TDT equipment

In particular, the linked lists are used often. They are used to store parameters regarding the protocol, the subject and any other elements of importance. Routines have been created which allow the linked lists to be read and stored to the disk (in configuration files).

4.1 Creating New Procedures

The general procedure for creating a new procedure will be outlined here. This is most likely the biggest addition to be performed with the software. The first and foremost thing to be considered is the requirements of the new procedure. It is up to the software designer to design a flexible procedure which can be used for many different experiments. It is most important that the designer consider all the possible parameters the experimenter may want to use. This is much easier to do at this stage compared to adding it at a later stage. Even if the parameter never gets used, it is best to add it at the beginning.

Each procedure is contained within its own source file. It is recommended to put all relevant software within the one file. The makefile and link file need to be changed to include the new source file. To incorporate the software into the procedures available to the administrator (and thus the experimenter), a new record must be added to the 'protocol.ini' file in the /config area. This file defines all the parameters and their possible values. The mainline will take care of presenting the administrator with the possible parameters and their corresponding choices.

The actual names of the parameters are arbitrarily chosen by the software designer. The new software routines must simply understand the parameters defined by the 'protocol.ini' file. In theory this is all that is required to create a new procedure. Obviously, it is up to the software designer to fully test and debug the new software routine.

The software designer should make heavy use of the existing routines (linked lists, menus, log files, etc...). Those routines are fully debugged and are known to work. They can greatly increase the speed at which new procedures can be developed. The software designer should only have to write the software pertaining to actually controlling the noise environment. The underlying routines to help do this are already written.

DOCUMENT CONTROL DATA SHEET	
1a. PERFORMING AGENCY Garry Dunn, Trellis Consulting, 9 Holly Meadow Road, Barrie ON L4N 8S1.	2. SECURITY CLASSIFICATION UNCLASSIFIED –
1b. PUBLISHING AGENCY DRDC Toronto	
3. TITLE Development and Documentation of Software to Control the Noise Simulation Facility at DRDC Toronto	
4. AUTHORS Garry Dunn	
5. DATE OF PUBLICATION February 15 , 2005	6. NO. OF PAGES 11
7. DESCRIPTIVE NOTES	
8. SPONSORING/MONITORING/CONTRACTING/TASKING AGENCY Sponsoring Agency: Monitoring Agency: DRDC Toronto Contracting Agency : DRDC Toronto Tasking Agency: DRDC Toronto	

9. ORIGINATORS DOCUMENT NO.	10. CONTRACT GRANT AND/OR PROJECT NO.	11. OTHER DOCUMENT NOS.
Contract Report CR 2005–083	16cb19	

12. DOCUMENT RELEASABILITY
Unlimited distribution
13. DOCUMENT ANNOUNCEMENT
Unlimited announcement

14. ABSTRACT

(U) This document discusses the software used to control the Noise Simulation Facility at DRDC Toronto. Contained within are areas discussing the use of the software by both the administrator of the system as well as the experimenters themselves. The final section discusses how a programmer would expand the system to include more procedures.

(U)

15. KEYWORDS, DESCRIPTORS or IDENTIFIERS

(U) noise simulation; software development